Theme 1
Software Processes

Introduction to Software Engineering and
Quality in Software Engineering

Rehearsal

Learning Goals

- Why is software engineering important?
- Who and what does it consist of?
- What are its main activities?
- What are the principles of software engineering?
- What ethics are involved?

Table of content

- Definitions of SE
- Criticality
- P^4
- SE Principles
- Ethics

What is Software Engineering?

Definition and Issues

[Bruegge]

“state of the art of developing quality software on time and within budget”

- Trade-off between perfection and physical constraints
  SE has to deal with real-world issues
- State of the art!
  “best practice” is a moving target, community decides on it
  life-long learning

Definition and Issues

[Parnas]

"multi-person construction of multi-version software"

- Team-work
  Scale issue ("program well" is not enough) and communication issue
- Successful software systems must evolve or perish
  Change is the norm, not the exception

Definition and Issues

[Sommerville]

“software engineering is different from other engineering disciplines”

- Not constrained by physical laws
  limit = human mind
- It is constrained by political forces
  balancing stake-holders

Definition and Issues

[IEEE]

“The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, the application of engineering to software.”

- it is not only what is produced that is important but also how it is produced
- we can expect the production of software that is reliable and maintainable and meets the requirements by applying an analogous set of approaches to the development of software.

Why is Software Engineering Critical?

Ariane 5

Destruction after 37 seconds flight on the
4th of June 1996 at Kourou

- Malfunction of the software
- One of the most expensive computer bugs in history

- problem is easy to describe, not easy to avoid
- extensive education and coordination within project management, quality assurance, configuration management, architecture, detailed design, programming and testing organisations.

Radiation Overdose

- software controls ever-increasing number of devices
- From project management magazine Baseline
  - “massive overdoses of gamma rays partly due to limitations of the computer program that guided use of” a particular radiation-therapy machine.

Why software fails

- over budget
- exceeds schedule and/or misses market window
- doesn’t meet stated customer requirements
- lower quality than expected
- performance does not meet expectations
- too difficult to use

P^4

- People
  - several groups of people involved with a project's outcome = *stakeholders*
  - *Business management*
    focus is on business issues including profit, cost effectiveness, customer satisfaction
  - *project management*
    responsible for planning and tracking a project
  - *development team*
    responsible for developing and maintaining the project
  - *customers*
    responsible for purchasing the software
  - *end users*

    interact with and use software after it is finished being developed.

P^4 (continued)

- Product
  - Project documentation = documents produced during software definition and development
  - Code = source and object
  - Test documents = plans, cases, and results
  - Customer documents = documents explaining how to use and operate product
  - Productivity measurements: analyze project productivity

P^4 (continued)

- Project
  - defines the activities and associated results needed to produce a software product. Activities:
    - Planning: plan, monitor and control the software project
    - Requirements analysis: define what to build
    - Design: describe how to build the software
    - Implementation: program the software
    - Testing: validate that software meets the requirements
  - various development paradigms, techniques and tools exist and are employed on different projects
    - e.g., the object-oriented paradigm

P^4 (continued)

- Process
  - Framework for carrying out the activities of a project in an organized and disciplined manner.
    - imposes structure and helps guide people and activities
    - expresses the interrelationship among the phases by defining their order and frequency, and by defining the deliverables of the project.

Software Engineering Principles

1. Make quality number 1
2. High-quality software is possible
3. Give products to customers early
4. Use an appropriate software process
5. Minimize intellectual distance
6. Inspect code
7. People are the key to success

source: 201 Principles of Software Engineering, Alan Davis

Ethics in Software Engineering

- Most disciplines operate under a strict set of ethical standards

- The Merriam-Webster online dictionary defines ethics as:

  1. the discipline dealing with what is good and bad and with moral duty and obligation
  2. a set of moral principles

PREAMBLE

The short version of the code summarizes aspirations at a high level of the abstraction; the clauses that are included in the full version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.

Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

1. PUBLIC - Software engineers shall act consistently with the public interest.

2. CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.

3. PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

4. JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.

5. MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

6. PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

7. COLLEAGUES - Software engineers shall be fair to and supportive of their coleagues.

8. SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Case Studies

- Gaining thorough understanding of software engineering concepts through:
  - learn how they are applied to the development of real software applications
    Interactive sessions, case study: Pay Station Case
  - gain hands-on experience developing a software application as part of a team.
    Project!!!